

Fast Approximation and Randomized Algorithms for Diameter

Sharareh Alipour* and Bahman Kalantari† and Hamid Homapour‡

Abstract

We consider approximation of diameter of a set S of n points in dimension m . Egecioğlu and Kalantari [6] have shown that given any $p \in S$, by computing its farthest in S , say q , and in turn the farthest point of q , say q' , we have $\text{diam}(S) \leq \sqrt{3} d(q, q')$. Furthermore, iteratively replacing p with an appropriately selected point on the line segment pq , in at most $t \leq n$ additional iterations, the constant bound factor is improved to $c_* = \sqrt{5 - 2\sqrt{3}} \approx 1.24$. Here we prove when $m = 2$, $t = 1$. This suggests in practice a few iterations may produce good solutions in any dimension. Here we also propose a randomized version and present large scale computational results with these algorithm for arbitrary m . The algorithms outperform many existing algorithms. On sets of data as large as 1,000,000 points, the proposed algorithms compute solutions to within an absolute error of 10^{-4} .

Keywords: Diameter, Approximation Algorithms, Randomized Algorithms

1 Introduction

Given a finite set of points S in \mathbb{R}^m , the diameter of S , denoted by $\text{diam}(S)$, is defined as the maximum distance between two points of S . Yao [13] has considered the case of $m > 2$. For $m = 2$ the problem can be solved in $O(n \log n)$ time. Computing the diameter of a point set is a fundamental problem and has a long history. It can be shown that computing the diameter of n points in \mathbb{R}^m requires $\Omega(n \log n)$ operations in the algebraic computation-tree model [10]. The problem becomes much harder in \mathbb{R}^3 . Clarkson and Shor gave a randomized $O(n \log n)$ algorithm [4]. Recent attempts to solve the 3-dimensional diameter problem led to $O(n \log^3 n)$ [11, 1] and $O(n \log^2 n)$ deterministic algorithms [11, 3]. Finally Ramos found an optimal $O(n \log n)$ deterministic algorithm [12]. All these algorithms use complex data structures and algorithmic techniques such as 3-dimensional convex hulls, intersection of balls, furthest-point Voronoi diagrams, point location search structures or parametric search. There are many other papers that focus on this problem, see [7, 2, 5, 9, 8]. The first nontrivial approximation algorithm for this problem for arbitrary m was given in [6], approximating the diameter to within a factor of $\sqrt{3}$. The operation cost of this algorithm is $O(mn)$. Additionally, [6] describes an iterative algorithm that in $t \leq n$ iterations, each of cost $O(mn)$, produces an approximation of $\text{diam}(S)$ to within a factor of $c_* = \sqrt{5 - 2\sqrt{3}} \approx 1.24$.

In this paper we first prove that for $m = 2$ it is possible to produce an approximation of diameter to within the factor of c_* in $t = 2$ iterations, thus giving an $O(n)$ approximation algorithm. In fact running this algorithm for general case of m only $t = 2$ iterations produces very good approximation for large test data. Additionally, we describe a simple randomized algorithm to approximate the diameter of a finite set of points in any dimension m . This algorithm is a modified version of the algorithm presented in [6]. We also test this algorithm for large data sets, making comparison with several algorithms in the literature. Our computational results demonstrate that the proposed algorithms here are superior in performance to the existing ones. The proposed algorithms appear to be extremely fast for a large variety of point distributions, in large dimensions. Moreover, these algorithms do not need to construct any complicated data structure and

*Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. shalipour@ce.sharif.edu, This work was completed in part when the author visited DIMACS as an REU mentor in 2014

†Department of Computer Science, Rutgers University, New Brunswick, New Jersey, USA. kalantari@cs.rutgers.edu

‡Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. homapour@ce.sharif.edu

very easy to implement. In addition to the memory required for the data, they only use constant memory. The most relevant work to ours are those in [9, 8] which we make comparison to.

In Section 2, we present an approximation algorithm described in [6] but prove that in 2D it approximated the diameter to within a factor of $c_* = \sqrt{5 - 2\sqrt{3}} \approx 1.24$. In Section 3, we formally describe this algorithm for arbitrary dimension m , and give a randomized version. In Section 4, we present experimental results of the proposed algorithms in various dimensions and make comparison with several existing algorithms.

2 A Fast approximation of Diameter in 2D

Let $S = \{p_1, \dots, p_n\}$ be a subset of \mathbb{R}^m . We will first assume $m \geq 2$ is an arbitrary integer and describe an approximation algorithm but we will analyze the performance for $m = 2$. Let $\text{diam}(S)$ be the diameter of S . Let $d(\cdot, \cdot)$ denote the Euclidean distance. Given $p \in \mathbb{R}^m$, $r > 0$, let $B_r(p) = \{x \in \mathbb{R}^m : d(x, p) \leq r\}$, the ball of radius r centered at p . For a given point $p \in \mathbb{R}^m$, let $f(p)$ denote the farthest point of p in S . Let $r_p = d(p, f(p))$. We write $f^2(p)$ for $f(f(p))$.

Consider the following algorithm. Pick arbitrary $p \in S$. Compute $f(p)$. Clearly, $S \subset B_{r_p}(p)$, see Figure 1, and we have,

$$r_p \leq \text{diam}(S) \leq 2r_p. \quad (1)$$

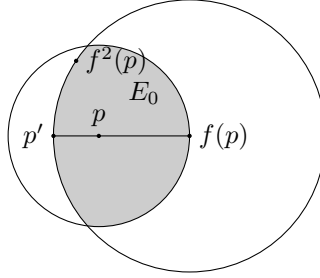


Figure 1: E_0 , an initial region containing S based on two farthest point computations.

Next compute $f^2(p)$. Let

$$E_0 = B_{r_p}(p) \cap B_{r_{f(p)}}(f(p)). \quad (2)$$

Clearly, $S \subset E_0$. The set E_0 is the intersection of two balls, forming an uneven eye-shape, see gray area in Figure 1. Its diameter gives a better factor bound than 2. To estimate the diameter of E_0 we include it in a larger eye-shape region whose diameter can be estimated conveniently. Set

$$p' = \alpha p + (1 - \alpha)f(p), \quad \alpha = \frac{r_{f(p)}}{r_p}. \quad (3)$$

This point p' lies on the affine line joining p and $f(p)$, a distance of $r_{f(p)}$ from $f(p)$. Let

$$E = B_{r_{f(p)}}(p') \cap B_{r_{f(p)}}(f(p)). \quad (4)$$

The eye-shape region E is the intersection of two balls passing through each other's centers, see gray area in Figure 2. The diameter of E is known to be $\sqrt{3}r_{f(p)}$, see [6]. From this it follows that

$$r_{f(p)} \leq \text{diam}(S) \leq \sqrt{3}r_{f(p)}. \quad (5)$$

The diameter of E is attained as the distance between the two corners of this eye-shape. Let c_1 and c_2 be these corners, see Figure 2. Clearly the complexity to obtain this $\sqrt{3}$ -approximation to diameter is $2mn$ arithmetic operations.

To improve this bound, in [6] the following iterative procedure is described: Let q be the midpoint of p' and $f(p)$, see Figure 3 for a case in the Euclidean plane. Specifically, from 3 we have

$$q = \frac{1}{2}p' + \frac{1}{2}f(p) = \frac{\alpha}{2}p + (1 - \frac{\alpha}{2})f(p), \quad \alpha = \frac{r_{f(p)}}{r_p}. \quad (6)$$

Compute $f(q)$ and $f^2(q)$. If $d(f(q), f^2(q)) \leq d(f(p), f^2(p))$, then

$$\text{diam}(S) \leq c_* d(f(p), f^2(p)), \quad c_* = \sqrt{5 - 2\sqrt{3}} \approx 1.24. \quad (7)$$

Otherwise, replaces S with $S \setminus \{p, f(p)\}$, and repeat the process, replacing p with q , $f(p)$ with $f(q)$. That is, let q' be the point on the line segment $qf(q)$ a distance of $r_{f(q)}/2$ from $f(q)$. Then compute $f(q')$ and $f^2(q')$ and checks if $d(f(q'), f^2(q')) \leq d(f(q), f^2(q))$. If so, $\text{diam}(S) \leq c_* d(f(q), f^2(q))$. If not, iterates again. Eventually, in $t \leq n$ iterations, each of cost $O(mn)$, we obtain an approximation of $\text{diam}(S)$ to within a factor of c_* . However, in [6] no constant bound on t is given. In the forgoing arguments we prove that when $m = 2$,

$$\text{diam}(S) \leq c_* \max \left\{ d(f(p), f^2(p)), d(f(q), f^2(q)) \right\}. \quad (8)$$

Hence in at most $8n$ operations, the cost of computing the farthest point of 4 points, namely $f(p)$, $f^2(p)$, $f(q)$, $f^2(q)$, we have an approximation of diameter to within a factor of c_* . We thus improve the results in [6] for $m = 2$.

We now proceed to prove this result. Having picked an arbitrary point $p \in S$, we compute $f(p)$ and $f^2(p)$, and p' as described above. Let q be the midpoint of p' and $f(p)$, see Figure 3 for a case in the Euclidean plane. Compute $f(q)$. Let

$$\rho_* = \frac{c_*}{2} = \frac{\sqrt{5 - 2\sqrt{3}}}{2} \approx .62. \quad (9)$$

If

$$r_q \leq \rho_* r_{f(p)}, \quad (10)$$

then

$$\text{diam}(S) \leq 2r_q \leq c_* r_{f(p)}. \quad (11)$$

This is obvious since $r_q \leq \rho_* r_{f(p)}$ implies S is contained in $B_{r_q}(q)$. So we assume $r_q > \rho_* r_{f(p)}$.

Without loss of generality assume that

$$p' = (-\frac{1}{2}, 0, \dots, 0) \in \mathbb{R}^m, \quad f(p) = (\frac{1}{2}, 0, \dots, 0) \in \mathbb{R}^m. \quad (12)$$

Thus $q = 0$ is the midpoint of the line segment $p'f(p)$. We may also assume that the corner points of E are located at

$$c_1 = (0, \dots, 0, \frac{\sqrt{3}}{2}) \in \mathbb{R}^m, \quad c_2 = (0, \dots, 0, \frac{\sqrt{3}}{2}) \in \mathbb{R}^m. \quad (13)$$

Let E_1 and E_2 be defined as the two halves of E determined as the intersection of E with the orthogonal bisecting hyperplane to the line c_1c_2 (the upper and lower parts of E). We thus have $r_{f(p)} = 1$. Let

$$r = r_q = d(q, f(q)). \quad (14)$$

We assume $r > \rho_*$ (since otherwise $\text{diam}(S) \leq c_*$). To improve the bound on diameter we compute of $f^2(q)$, the farthest point of $f(q)$ in S . Let

$$d = d(f(q), f^2(q)). \quad (15)$$

We first prove that if $d \geq 1$ the following holds

$$d \leq \text{diam}(S) \leq c_* d. \quad (16)$$

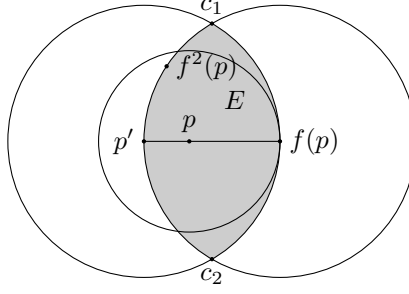


Figure 2: E , a region that contains S with diameter bounded by $\sqrt{3}\text{diam}(S)$.

To prove this, consider first the case of Euclidean plane, see Figure 3. Assume without loss of generality $f(q)$ has nonnegative coordinates.

Let $q_* = (x_*, y_*)$ be the intersection of the two circles

$$(x - \frac{1}{2})^2 + y^2 = 1, \quad x^2 + y^2 = r^2, \quad (17)$$

where $x_* < 0$, and $y_* > 0$. This gives

$$x_* = r^2 - \frac{3}{4} < 0. \quad (18)$$

We have

$$y_*^2 = r^2 - x_*^2 = r^2 - (r^2 - \frac{3}{4})^2 = -r^4 + \frac{5}{2}r^2 - \frac{9}{16}. \quad (19)$$

So

$$y_* = \sqrt{-r^4 + \frac{5}{2}r^2 - \frac{9}{16}}. \quad (20)$$

Let w be the solution to the intersection of the sphere of radius d centered at $f(q)$ and the sphere $(x + \frac{1}{2})^2 + y^2 = 1$. Thus

$$d(f(q), w) = d. \quad (21)$$

Note that we must have $d(q, w) \leq d(q, f(q))$. Let $\bar{q}_* = (-x_*, y_*)$. Let $q'_* = (u, v)$ be the solution to

$$(x + \frac{1}{2})^2 + y^2 = 1, \quad (x + x_*)^2 + (y - y_*)^2 = d^2, \quad (22)$$

where $v \leq 0$, (see Figure 3). Thus

$$d = d(\bar{q}_*, q'_*). \quad (23)$$

Let $r' = d(q, q'_*)$. Thus $u^2 + v^2 = r'^2$.

Let

$$d_* = d(q_*, q'_*), \quad d' = d(q_*, w). \quad (24)$$

We first prove some lemmas.

Lemma 1. *Suppose $r' \leq r$ and $d \geq 1$. Then $\text{diam}(E \cap B_r(q) \cap B_d(f(q))) = d'$.*

Proof. Consider a pair of points u, v in the corresponding domain such that $\text{diam}(E \cap B_r(q) \cap B_d(f(q))) = d(u, v)$, see Figure 3. We claim that u and v must be the extreme points of the domain (corner points). Otherwise, assuming that u, v do not coincide with $p', f(p)$, consider the line segment uv . Then the line that is orthogonal to this line segment, either at u or at v , must cut through the feasible region. But this means

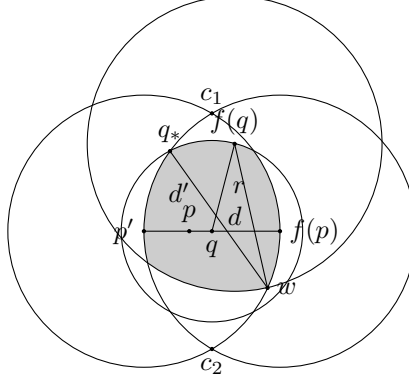


Figure 3: The region $E \cap B_r(q) \cap B_d(f(q))$, having diameter d' .

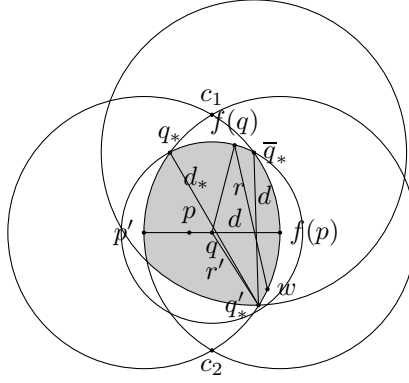


Figure 4: The region $E \cap B_r(q) \cap B_d(\bar{q}_*)$, having diameter d_* .

$d(u, v)$ is not the claimed diameter since it can be increased within the feasible domain. On the other hand if u, v does coincide with $p', f(p)$, then we can change $f(p)$ to a corner point on its bounding circle until it touches another circle in which case we can replace it with a corner point. Now once one of u or v is a corner point, say u , we can consider the orthogonal line to uv at v . Again this line must cut through the feasible region, contradicting that $d(u, v)$ is diameter. Once we have argued that both u, v are extreme points we can consider different pairs and since we have assumed that $f(q)$ has nonnegative coordinates it follows that the diameter is $d' = d(q_*, w)$ as claimed. \square

Lemma 2. Suppose $r' \leq r$ and $d \geq 1$. Then $\text{diam}(E \cap B_{r'}(q) \cap B_d(\bar{q}_*)) = d_*$.

Proof. For this consider Figure 4. The proof of this lemma is analogous to the proof of previous one. \square

Lemma 3. Suppose $r' \leq r$ and $d \geq 1$. Then $d \leq \text{diam}(S) \leq d' \leq d_*$.

Proof. The first two inequalities are obvious. To prove the last inequality we only need to observe that w is feasible to $E \cap B_d(\bar{q}_*)$ (even though $E \cap B_{r'}(q) \cap B_d(\bar{q}_*)$ is not a subset of $E \cap B_r(q) \cap B_d(f(q))$). Figure 5 gives a superposition of the regions. \square

Next we state our main result.

Theorem 1. *Suppose $r' \leq r$ and $d \geq 1$. Then, $d_* \leq c_* d$.*

Proof. Note that

$$d_*^2 = (u - x_*)^2 + (v - y_*)^2. \quad (25)$$

Also $u^2 + v^2 = r'^2$. From the first equation in (22) we get

$$u = \frac{3}{4} - r'^2. \quad (26)$$

Note that

$$v^2 = r'^2 - u^2 = r'^2 - \left(\frac{3}{4} - r'^2\right)^2 = -r'^4 + \frac{5}{2}r'^2 - \frac{9}{16}. \quad (27)$$

So

$$v = -\sqrt{-r'^4 + \frac{5}{2}r'^2 - \frac{9}{16}}. \quad (28)$$

Then from the second equation in (22) we get

$$(u^2 + v^2) + (x_*^2 + y_*^2) + 2ux_* - 2vy_* = d^2. \quad (29)$$

Since $u^2 + v^2 = r'^2$ and $x_*^2 + y_*^2 = r^2$ we get

$$2(x_*u - y_*v) = d^2 - r^2 - r'^2. \quad (30)$$

We wish to bound the ratio d_*/d where

$$d_*^2 = (u - x_*)^2 + (v - y_*)^2. \quad (31)$$

Note that we have

$$d_*^2 = r^2 + r'^2 - 2x_*u - 2y_*v, \quad d^2 = r^2 + r'^2 + 2x_*u - 2y_*v. \quad (32)$$

So we get

$$d_*^2 = d^2 - 4x_*u. \quad (33)$$

Substituting for x_* and u and dividing by d^2 we get

$$\frac{d_*^2}{d^2} = 1 + \frac{4}{d^2} \left(\frac{3}{4} - r^2\right) \left(\frac{3}{4} - r'^2\right). \quad (34)$$

Note that

$$d^2 = r'^2 + r^2 - 2\left(\frac{3}{4} - r^2\right)\left(\frac{3}{4} - r'^2\right) + 2\sqrt{-r'^4 + \frac{5}{2}r'^2 - \frac{9}{16}}\sqrt{-r'^4 + \frac{5}{2}r'^2 - \frac{9}{16}}. \quad (35)$$

Let $a = r^2$, $b = r'^2$. Since $r' \leq r$, we wish to bound the following ratio in the region $1/4 < b \leq a \leq 3/4$:

$$\frac{4\left(\frac{3}{4} - a\right)\left(\frac{3}{4} - b\right)}{a + b - 2\left(\frac{3}{4} - a\right)\left(\frac{3}{4} - b\right) + 2\sqrt{-a^2 + \frac{5}{2}a - \frac{9}{16}}\sqrt{-b^2 + \frac{5}{2}b - \frac{9}{16}}}. \quad (36)$$

Multiplying and dividing by 4 and simplifying we get

$$\frac{(3 - 4a)(3 - 4b)}{4a + 4b - 0.5(3 - 4a)(3 - 4b) + 0.5\sqrt{-16a^2 + 40a - 9}\sqrt{-16b^2 + 40b - 9}}. \quad (37)$$

Let $A = 4a$ and $B = 4b$. Then the ratio becomes

$$F(A, B) = \frac{(3 - A)(3 - B)}{A + B - 0.5(3 - A)(3 - B) + 0.5\sqrt{-A^2 + 10A - 9}\sqrt{-B^2 + 10B - 9}}, \quad 1 < B \leq A \leq 3. \quad (38)$$

When $A = B$ we get

$$R(A) = F(A, A) = \frac{(3-A)^2}{2A - 0.5(3-A)^2 + 0.5(-A^2 + 10A - 9)} = \frac{(3-A)^2}{-A^2 + 10A - 9}, \quad 1 < A \leq 3. \quad (39)$$

The function $R(A)$ is monotonically decreasing on $[1 + \epsilon, 3]$, $\epsilon > 0$ arbitrarily small. Using this, we can find the appropriate bound. So it suffices to show that for fixed B , $1 < B < A$, we have $F(A, B) \leq R(A)$. But this can be verified by calculus (also graphing it shows this).

Since $d_*/d \leq \sqrt{1 + R(A)}$, and $1 < A = 4r^2 \leq 3$, we find a value for r so that

$$2r = \sqrt{1 + R(A)} = \sqrt{1 + \frac{(3-A)^2}{-A^2 + 10A - 9}}. \quad (40)$$

Equivalently, squaring the above we get

$$A = 1 + R(A). \quad (41)$$

Solving this we get $A = 5 - 2\sqrt{3} \approx 1.5359$. Then $r = \sqrt{5 - 2\sqrt{3}}/2 \approx .62$, and

$$\sqrt{1 + R(A)} = \sqrt{5 - 2\sqrt{3}} \approx 1.24. \quad (42)$$

□

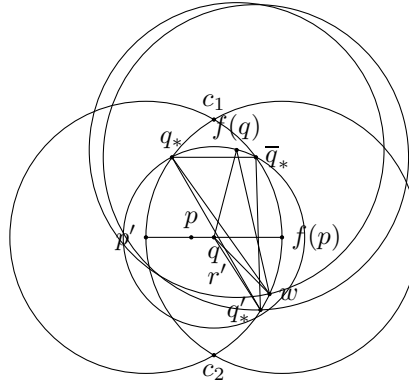


Figure 5: Superposition of E , $B_r(q)$, $B_d(f(q))$, and $B_d(\bar{q}_*)$.

Theorem 2. Suppose $r' > r$ and $d \geq 1$. Then, $2y_* \leq d \leq \text{diam}(S) \leq 2r$. In particular, $\text{diam}(S) \leq c_* d$.

Proof. Since $r' > r$, q'_* is outside of the ball of radius r at q . This means $d \geq 2y_*$ (see definition of y_* , (19)). In other words d is at least as long as the distance between \bar{q}_* and its reflection with respect to the x -axis. Now given the formula of y_* in terms of r , (19) we have

$$\frac{r}{y_*} = \frac{r}{-r^4 + \frac{5}{2}r^2 - \frac{9}{16}}. \quad (43)$$

As a function of r the maximum in the interval $[\rho_*, \sqrt{3}/2]$ is attained at ρ_* and is c_* . It decreases to one at the other end point of the interval. □

We now prove the result when $d \leq 1$.

Theorem 3. Suppose $d \leq 1$. Then, $\text{diam}(S) \leq c_*$.

Proof. Since $S \subset E \cap B_r(q) \cap B_d(f(q)) \subset E \cap B_r(q) \cap B_1(f(q))$ and we have already proved the diameter of $E \cap B_r(q) \cap B_1(f(q))$ is bounded above by c_* , the proof of the Euclidean plane is complete. \square

We end this analysis by exhibiting a case of five points where the worst-case bound of c_* is achievable, see Figure 6.

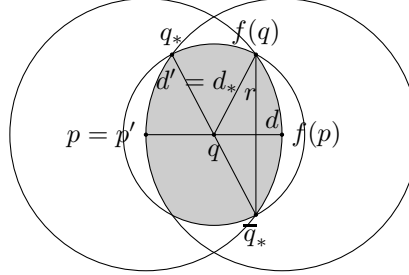


Figure 6: A case of five points gives worst-case error when $r = \rho_*$.

3 Iterative Algorithms

In this section we propose two iterative algorithms for approximating the diameter of a finite set in any dimension $m \geq 2$. The first algorithm is essentially identical with the algorithm in [6]. The second is a randomized version of the algorithm. We formally describe these in Algorithm 1 and Algorithm 2. They each have an input t as the number of iterations, however we only give implementation of them for small t because they produce high accuracy solution (within absolute error of 10^{-4} on tested data sets). We present experimental results with these algorithms.

3.1 The iterative approximation algorithm

In this section an iterative version of previous algorithm is presented. In each iteration given $p \in S$, we compute $f(p)$, $f^2(p)$, q , $f(q)$ and $f^2(q)$. Computing $f(p)$ and $f^2(p)$ requires at most $2mn$ operations. Computing q requires $O(m)$ operations (see 6). An additional $2mn$ operations are needed to compute $f(q)$ and $f^2(q)$. Thus when m is much smaller than n each iteration requires $4mn$ operations. In our computational results we have only run the algorithm $t = 2$ times. In Algorithm 1, the iterative algorithm is shown formally.

3.2 Randomized and approximation Algorithm

In this section, we present a randomized version of Algorithm 1. We begin from an arbitrary point p and compute $f(p)$. Next we compute the midpoint of p and $f(p)$. Let q be this midpoint. Then we compute $f(q)$ and $f^2(q)$. Thus the maximum of $d(p, f(p))$, $d(q, f(q))$, $d(f(q), f^2(q))$ is a lower bound to the diameter of S . We iterate this algorithm. In the next step, we can either begin from $f(p)$ or $f^2(q)$. To do so, we randomly choose $f(p)$ or $f^2(q)$ with equal probability. This becomes our new point. Then we compute the farthest point from the chosen point and compare the estimate of diameter of previous step with the new one. In Algorithm 2 we explain the algorithm formally. In each iteration of the algorithm we need about $3mn$ operations in contrast with $4mn$ operations in Algorithm 2. In practice we ran this with $t = 2, 3$ and 5. We have implemented this algorithm on some data sets. In the next section the experimental results of this algorithm are shown.

Algorithm 1 The iterative approximation algorithm

Input:

S : a set of n points in \mathbb{R}^m

t : number of iterations

Output:

An approximation value of $\text{diam}(S)$

$d_{max} = 0, i = 0.$

choose an arbitrary point $p \in S$.

while $i < t$ **do**

if $r_p = d(p, f(p)) > d_{max}$ **then**

$d_{max} = d(p, f(p))$

end if

if $r_{f(p)} = d(f(p), f^2(p)) > d_{max}$ **then**

$d_{max} = d(f(p), f^2(p))$

end if

 Let $q = \frac{\alpha}{2}p + (1 - \frac{\alpha}{2})f(p)$, $\alpha = \frac{r_{f(p)}}{r_p}$.

if $d(f(q), f^2(q)) > d_{max}$ **then**

$d_{max} = d(f(q), f^2(q))$

end if

$i = i + 1$

$p = f^2(q)$

end while

return d_{max}

4 Experimental results

To show the efficiency of the proposed algorithms in practice, we have implemented these and run them on some data sets. The most comparable approaches to ours are the algorithms proposed in [9] and [8]. We have used the package implemented by Malandain and Boissonnat's in [9]. They have implemented their algorithms and we have also implemented our algorithms and added them to their package. In their experiments they generated 2 types of data set: Volume based distributions, in a cube, in a ball, and in sets of constant width (only in 2D); and Surface based distributions, on a sphere, and on ellipsoids. They also used real inputs¹. We have also used the same package to generate data sets and the same real inputs. Malandain and Boissonnat's have implemented the following algorithms in [9]:

- Malandain and Boissonnat's exact algorithm;
- Malandain and Boissonnat's approximation algorithm;
- Har-Peled's algorithm: implemented by Malandain and Boissonnat;
- Hybrid1 algorithm: proposed by Malandain and Boissonnat which is combination of their algorithm and Har-Peled's algorithm;
- Hybrid2 algorithm: another modification of the two algorithms presented by Malandain and Boissonnat's algorithm and Har-Peled's algorithm presented by Malandain and Boissonnat's.

We have generated the data sets and computed the diameter for each set using each of the above algorithms and our proposed algorithms. The experimental results are shown in the tables. The first 5 algorithms are implemented by Malandain and Boissonnat's and the next one is the implementation of Algorithm 1 with $t = 2$ iterations. The next 3 are implementation of Algorithm 2 with $t = 2, 3$ and 5 iterations.

We make some observations in our computation. Firstly, in all the data sets, the difference between the approximated value and exact value of diameter is less than 10^{-4} where $\text{diam}(S) > 1$ even with $t = 2$

¹ Large Geometric Models Archive, http://www.cs.gatech.edu/projects/large_models/, Georgia Institute of Technology.

Algorithm 2 The iterative randomized algorithm

Input: S : a set of n points in \mathbb{R}^m t : number of iterations*Output:*An approximation value of $\text{diam}(S)$ $d_{max} = 0, i = 0.$ choose an arbitrary point $p \in S$.**while** $i < t$ **do** **if** $d(p, f(p)) > d_{max}$ **then** $d_{max} = d(p, f(p))$ **end if** Let q be the middle point on the line connecting p and $f(p)$ **if** $d(f(q), f^2(q)) > d_{max}$ **then** $d_{max} = d(f(q), f^2(q))$ **end if** $i = i + 1$ with probability 0.5 let $p = f(p)$ and with probability 0.5 let $p = f^2(q)$ **end while****return** d_{max}

iterations for both algorithms. From the tables it is seen that the running time of the randomized algorithm, Algorithm 2 with $t = 2$ iteration is better than all the other algorithms. The proposed algorithms are more efficient in higher dimensions. One advantage of the proposed algorithms is that no extra memory is needed. Also, by virtue of their efficiency these algorithms can be implemented for big data sets. Another advantage of these algorithms is that in higher dimensions, the running time of these algorithm is significantly better than the other algorithms (see Table 3).

Table 1: CPU times in millisecond for real data sets

Input	blade	dragon	hand	happy
Exact	70.9	8.06	17	12.38
Approx	67	5.9	15	6.27
Har-Peled	84	84.6	36.8	76
Hybrid1	112	102	13	64.2
Hybrid2	42	79	13	54
Algorithm 1	25.4	12.2	8.31	14.3
Algorithm 2-5	34	16.9	14.2	20.5
Algorithm 2-3	20.4	10.2	7.94	12.3
Algorithm 2-2	1.35	6.9	5.3	8.2

5 Conclusion

In this paper, we studied computing the diameter of a point set in any dimension which is a significant problem in computational geometry. We presented a fast constant approximation factor algorithm, giving a worst-case bound of about 1.24 in dimension $m = 2$. We believe that the same bound applies to any dimension. Its verification is the subject of future work. We proposed two iterative algorithms, one a randomized iterative algorithm. We also implemented these algorithms and compared the running times with related works. Based on experimental results the algorithms are very efficient. Deriving worst-case bounds for iterative algorithms in terms of the number of iterations t remains as open problem.

Table 2: CPU time in milliseconds 3D Volume and Surface Based distributions

Input Points	Cube 10,000	Cube 100,000	Cube 1,000,000	Ball 10,000	Ball 100,000	Ball 200,000
Exact	1.2	6	72	0.23	1.88	7.43
Approx	0.3	0.7	41	0.2	1.97	3.9
Har-Peled	2	16	322	0.39	10	41.70
Hybrid1	2	17	183	1.5	16.9	63.11
Hybrid2	1.94	16	163	1.4	27.4	62.11
Algorithm 1	0.25	3.1	30.9	0.32	2.68	5.3
Algorithm 2-5	0.34	3.6	38	0.52	3.981	7.574
Algorithm 2-3	0.21	2.31	26.32	0.214	2.19	4.557
Algorithm 2-2	0.18	1.49	17.48	0.148	1.48	3.08
Input Points	Sphere 10,000	Sphere 100,000	Sphere 200,000	Ellipsoid 1,000,000	Ellipsoid rotated 1,000,000	Ellipsoid regular 1,000,000
Exact	0.29	3.03	5.9	100.28	52.81	51.86
Approx	0.27	2.8	5.7	106.08	51.92	52.53
Har-Peled	122.71	7.96	33.27	284.69	103.88	113.8
Hybrid1	1.29	0.75	3.22	95.19	66.65	66.93
Hybrid2	1.43	0.71	2.94	0.114	65.1	68.75
Algorithm 1	0.27	2.6	6.1	23.5	26.84	27.8
Algorithm 2-5	0.34	3.59	7.69	44.3	38.65	38.89
Algorithm 2-3	0.23	2.17	4.56	25.89	23.91	23.39
Algorithm 2-2	0.14	1.53	3.66	15.23	15.77	15.44

Table 3: CPU times in millisecond for synthetic distributions in higher dimensions.

Cube, n=100,000				
Input	d=6	d=9	d=12	d=15
Exact	73.24	235.57	1520	27520
Approx	64.63	162.47	3210	16990
Har-Peled	207.52	24020	119890	138240
Hybrid1	87.23	180.31	17960	15450
Hybrid2	87.59	171.43	11800	12230
Algorithm 1	5.3	7.2	8.5	10.9
Algorithm 2-5	7.33	9.62	12.14	14.55
Algorithm 2-3	4.41	5.77	7.34	8.99
Algorithm 2-2	2.99	3.89	4.85	5.9
Ball, n=100,000				
Input	d=6	d=9	d=12	d=15
Exact	68770	111590	178430	264660
Approx	18550	48600	105200	168390
Har-Peled	86680	102800	120200	137530
Hybrid1	93940	136260	73620	222960
Hybrid2	71350	100030	146520	218510
Algorithm 1	5.5	6.42	8.12	9.46
Algorithm 2-5	7.25	9.58	11.98	14.35
Algorithm 2-3	4.38	5.87	7.29	8.64
Algorithm 2-2	2.89	3.89	4.82	5.79
Ellipse, n=100,000				
Input	d=6	d=9	d=12	d=15
Exact	58	1110	770	2760
Approx	31.9	504	589	1300
Har-Peled	3040	109760	127280	134070
Hybrid1	90	2280	3240	5580
Hybrid2	98	2020	3900	5200
Algorithm 1	5.21	6.9	7.28	9.5
Algorithm 2-5	7.3	9.5	12.1	16.5
Algorithm 2-3	4.39	5.79	7.2	8.6
Algorithm 2-2	2.988	4.2	4.9	5.8
Regular Ellipse, n=100,000				
Input	d=6	d=9	d=12	d=15
Exact	24	350	150	2280
Approx	31.7	285	110.5	1440
Har-Peled	530	62010	123720	133850
Hybrid1	71	308	3430	4070
Hybrid2	72.9	268	4590	119.5
Algorithm 1	5.06	6.84	8.22	10.7
Algorithm 2-5	7.3	11.2	12.2	17.8
Algorithm 2-3	4.39	5.95	7.3	8.65
Algorithm 2-2	2.952	4.05	4.8	5.8

References

- [1] Amato, Nancy M., Michael T. Goodrich, and Edgar A. Ramos. "Parallel algorithms for higher-dimensional convex hulls." *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*. IEEE, 1994.
- [2] Agarwal, Pankaj K., Shankar Krishnan, Nabil H. Mustafa, and Suresh Venkatasubramanian. "Streaming geometric optimization using graphics hardware." *Algorithms-ESA 2003*. Springer Berlin Heidelberg, 2003. 544-555.
- [3] Bespamyatnikh, Sergei. "An efficient algorithm for the three-dimensional diameter problem." *Discrete and Computational Geometry* 25.2 (2001): 235-255.
- [4] Clarkson, Kenneth L., and Peter W. Shor. "Applications of random sampling in computational geometry, II." *Discrete and Computational Geometry* 4.1 (1989): 387-421.
- [5] Chan, Timothy M. "Faster core-set constructions and data stream algorithms in fixed dimensions." *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004.
- [6] Egecioglu, Ömer, and Bahman Kalantari. "Approximating the diameter of a set of points in the Euclidean space." *Information Processing Letters* 32.4 (1989): 205-211.
- [7] Fournier, Herv, and Antoine Vigneron. "A tight lower bound for computing the diameter of a 3D convex polytope." *Algorithmica* 49.3 (2007): 245-257.
- [8] Har-Peled, Sariel. "A practical approach for computing the diameter of a point set." *Proceedings of the seventeenth annual symposium on Computational geometry*. ACM, 2001.
- [9] Malandain, Grgoire, and Jean-Daniel Boissonnat. "Computing the diameter of a point set." *International Journal of Computational Geometry and Applications* 12.06 (2002): 489-509.
- [10] Preparat, Franco P., and Michael Ian Shamos. "Computational geometry: an introduction." (1985).
- [11] Ramos, Edgar A. "Intersection of unit-balls and diameter of a point set in \mathbb{R}^d ." *Computational Geometry* 8.2 (1997): 57-65.
- [12] Ramos, Edgar A. "Deterministic algorithms for 3-D diameter and some 2-D lower envelopes." *Proceedings of the sixteenth annual symposium on Computational geometry*. ACM, 2000.
- [13] Yao, Andrew Chi-Chih. "On constructing minimum spanning trees in k-dimensional spaces and related problems." *SIAM Journal on Computing* 11.4 (1982): 721-736.